IN THE CLAIMS:

1. (Original) A method in an application server for initiating inter-process communication between non-persistent application sessions, the method comprising:

determining whether a second party is available to receive a message established in an application session of a first party; and

based on the determined availability of the second party, generating a HTML page having instructions for a browser to notify the second party of a new application session for the second party so as to present the message to the second party.

2. (Original) The method of claim 1, wherein the generating step includes inserting a uniform resource locator (URL) within the HTML page causing the browser to request interruption of a present application session of the second party to create the new application session for the second party.

3. (Currently Amended) The method of claim 2, further comprising generating a new session identifier that specifies the new application session for the second party, wherein the URL includes the new session identifier for interrupting the ~~a~~ present session of the second party with the new application session.

4. (Original) The method of claim 3, further including initiating an application instance for execution of the new application session for the second party based on a server-side data record configured for storing a state of the new application session and selected based on the new session identifier, in response to receipt of the URL from the browser.

5. (Previously Presented) The method of claim 1, wherein the HTML page includes a prompt enabling the second party to respond to the message.

6. (Original) The method of claim 1, wherein the determining step includes accessing a registry locally accessible by the application server, and the method further

including updating the registry to indicate that the first party is available for messaging operations.

7. (Original) The method of claim 1, further including storing the message in a data store of the second party.

8. (Original) The method of claim 7, wherein storing of the message is performed in accordance with IMAP protocol.

9. (Original) The method of claim 1, further including accessing attribute information of the second party to determine whether the second party authorizes receipt of the message from the first party.

10. (Original) The method of claim 9, wherein the attribute accessing step includes accessing a database server according to LDAP protocol.

11. (Original) The method of claim 1, wherein the message is a voice message and the HTML page includes instructions for playing the voice message.

12. (Original) A method for inter-process communication between non-persistent application instances, the method comprising:

establishing a first non-persistent application instance serving a first party;

establishing a second non-persistent application instance serving a second party; and

generating, in the first application instance, an HTML page having instructions for a persistent browser instance, having received the HTML page, to initiate a new application session for the second party.

13. (Original) The method of claim 12, further including accessing, by at least one of the first and second application instances, a common resource over an IP network.

14. (Original) The method of claim 13, wherein the common resource is a registry, the method including accessing the registry to determine whether the second party is currently active in the second application instance.

15. (Original) The method of claim 13, wherein the common resource is a data store for storing attribute information of each of the first and second parties, the method including accessing the data store in accordance with LDAP protocol.

16. (Currently Amended) The method of claim 13, wherein the common resource is a messages store for storing messages for each of the first and second parties, the method including accessing the message ~~data~~ store in accordance with IMAP protocol.

17. (Original) The method of claim 12, wherein the first application instance is established in first application server and the second application instance is established in a second application server.

18. (Original) The method of claim 13, wherein the common resource is accessible via an application programming interface (API).

19. (Original) The method of claim 12, further including initiating an application instance for execution of the new application session for the second party based on a server-side data record configured for storing a state of the new application session and selected based on the new session identifier, in response to receipt of the HTML page from the browser.

20. (Original) An application server configured for executing a messaging application, the application server including:

an application runtime environment configured for dynamically generating, for a first party, a hypertext markup language (HTML) document having instructions for a

browser to notify a second party of a new application session for the second party, based on a determination that the second party is available to receive the HTML document, the application runtime environment being configured to access a common resource containing information regarding both the first and second parties.

21. (Previously Presented) The application server of claim 20, wherein the HTML document has instructions to interrupt a present application session of the second party to create the new application session for the second party.

22. (Previously Presented) The application server of claim 20, wherein the HTML document includes a prompt enabling the second party to respond to the message.

23. (Original) The application server of claim 20, wherein the common resource includes a registry and the application runtime environment is configured to access the registry and to update the registry to indicate that the first party is available for messaging operations.

24. (Original) The application server of claim 20, wherein the application runtime environment is configured to access the common resource via an application programming interface (API).

25. (Original) The application server of claim 20, wherein the common resource includes user attribute information stored in a database server in accordance with LDAP protocol and the application runtime environment is configured to access the database server.

26. (Original) The application server of claim 20, wherein the common resource includes a message store for storing the message in accordance with IMAP protocol and the application runtime environment is configured to access the message store.

27. (Original) The application server of claim 20, wherein the common resource includes a registry and the application runtime environment is configured to access the registry and to determine whether the second party is available to receive the message.

28. (Original) A computer readable medium having stored thereon sequences of instructions for initiating inter-process communication between non-persistent application sessions, the sequences of instructions including instructions for performing the steps of:

determining whether a second party is available to receive a message established in an application session of a first party; and

based on the determined availability of the second party, generating a HTML page having instructions for a persistent browser to notify the second party of a new application session for the second party so as to present the message to the second party.

29. (Original) The medium of claim 28, wherein the generating step includes inserting a uniform resource locator (URL) within the HTML page causing the browser to request interruption of a present application session of the second party to create the new application session for the second party.

30. (Original) The medium of claim 29, further comprising generating a new session identifier that specifies the new application session for the second party, wherein the URL includes the new session identifier for interrupting a present session of the second party with the new application session.

31. (Original) The medium of claim 30, further including initiating an application instance for execution of the new application session for the second party based on a server-side data record configured for storing a state of the new application session and selected based on the new session identifier, in response to receipt of the URL from the browser.

32. (Previously Presented) The medium of claim 28, wherein the HTML page includes a prompt enabling the second party to respond to the message.

33. (Original) The medium of claim 28, wherein the determining step includes accessing a registry locally accessible by the application server, and the method further including updating the registry to indicate that the first party is available for messaging operations.

34. (Original) The medium of claim 28, further including storing the message in a data store of the second party.

35. (Original) The medium of claim 34, wherein storing of the message is performed in accordance with IMAP protocol.

36. (Original) The medium of claim 28, further including accessing attribute information of the second party to determine whether the second party authorizes receipt of the message from the first party.

37. (Original) The medium of claim 36, wherein the attribute accessing step includes accessing a database server according to LDAP protocol.

38. (Original) The medium of claim 28, wherein the message is a voice message and the HTML page includes instructions for playing the voice message.

39. (Original) An application server configured for executing a messaging application, the application server including:

means for dynamically generating, for a first party, a hypertext markup language (HTML) document having instructions for a browser to notify a second party of a new application session for the second party so as to present a message from the first party to the second party, based on a determination that the second party is available to receive the message.

40. (Previously Presented) The application server of claim 39, wherein the HTML document has instructions to interrupt a present application session of the second party to create the new application session for the second party.

41. (Previously Presented) The application server of claim 39, wherein the HTML document includes a prompt enabling the second party to respond to the message.